Classification with Controlled Robustness in High-Resolution SAR Data

Wolfgang Middelmann

FGAN-FOM Research Institute for Optronics and Pattern Recognition, Germany
W.Middelmann@fom.fgan.de

Abstract. Ground target classification in high-resolution SAR data has become increasingly important over the years. Kernel machines like the Support Vector Machine (SVM) and the Relevance Vector Machine (RVM) afford a great chance to solve this problem. But it is not possible to customize these kernel machines. Therefore the main objective of this work has been the development of a mechanism that controls the classification quality versus the computational effort. The investigations have been carried out with usage of the MSTAR public target dataset. The result of this work is an extended RVM, the RVMG. A single parameter is controlling the robustness of the system. The spectrum varies from a machine 15 times faster and of 10% lower quality than the SVM, goes to a 5 times faster and equal quality machine, and ends with a machine a little bit faster than the SVM and of better quality than the Lagrangian Support Vector Machine (LSVM).

1. Introduction

Ground surveillance and automatic target recognition are important tasks in military applications. The importance of SAR data has grown in this sector over the last decades. To solve these tasks high-performance classifiers are required. Kernel machines like the Support Vector Machine (SVM), see [9, 5, 2], the Lagrangian Support Vector Machine (LSVM), see [3], and the Relevance Vector Machine (RVM), see [6, 7] afford a great chance to solve this problem. For these investigations we use the Moving and Stationary Target Acquisition and Recognition (MSTAR) public target dataset, see [10].

In previous investigations [4] we have analyzed the utilizability of the mentioned kernel classifiers for the MSTAR dataset. The result of our former examination was an assessment of the classifiers. We measured the classification quality and the number of Support Vectors (SVs) which is directly proportional to the computational effort of evaluating the test function. An important drawback of the existing methods could be identified. Therefore we have formulated: *The main future objective should be the development of a mechanism for controlling the classification quality versus the number of Support Vectors. It should be preferable to design a kernel machine that could be directly or indirectly customized by a control parameter.* In this paper we want to describe one possible solution of this controlling via a special kind of parameterized generator that is integrated directly into the RVM. It fortifies boundary regions in dependency of the parameter, i.e. it strengthens the robustness of the original class structure for the classification training. The result is a classifier with indirectly controlled trade-off between speed and quality.

In section two an introduction of the MSTAR dataset and the experimental setup is given. In section three we recapitulate some previous results of [4]. The extension of the RVM follows in section four. A general extensibility of the RVM is described and a first shot is done to yield an upper limit of high classification quality and low computational effort for the considered family of kernel machines. In section five we introduce the main result, the RVM with a special kind of generator. The experimental results are given in section six followed by the conclusion in the last section.

2. MSTAR Data

The MSTAR dataset consists of training and test data. The training dataset was taken under a depression angle of 17°, the test data under 15°. The vehicles taken into consideration are organized in the three classes BMP2, BTR70, and T72, see Table 1.

Table 1. MSTAR - 1622 training and 1365 test chips organized in three classes

Class	**Types**	**Train 17° (1622)**	**Test 15° (1365)**
BMP2	9563, 9566, C21	698 (233+232+233)	587 (195+196+196)
BTR70	C71	233	196
T72	132, 812, S7	691 (232+231+228)	582 (196+195+191)

We use the magnitude of the complex data (128x128 pixel per chip) for classification purposes. Neither further preprocessing has been done like using superresolution methods (e.g. CLEAN, see [8]), nor feature-enhanced SAR processing has been taken into account, see [1]. In this paper we are interested in pure classification quality and computational effort of evaluating the test function of the classifiers. The experimental setup described above is the same as in [4]. Therefore the investigations are in a straight forward manner of the previous work.

3. Preliminary Investigations

In previous investigations the three mentioned kernel machines SVM, LSVM, and RVM are used with the Radial Basis Function (RBF) kernel. This kernel is defined by

$$K(x_1, x_2) = \exp\left(- \| x_2 - x_1 \|_2^2 / \sigma\right) \qquad \textbf{(1)}$$

with kernel parameter $\sigma > 0$. Its main advantage is that the topology of the 2-norm Voronoi diagram is invariant under the kernel generating function $\Psi : X \to F,\ \Psi(x) = z$ that holds

$$K(x_1, x_2) = < \Psi(x_1), \Psi(x_2) > \qquad \textbf{(2)}$$

with $<.,.>$ the l_2 -inner product, for further details see [2].

The kernel classifiers are tested with the MSTAR dataset. The results of classification quality (in % of the 1365 test chips) and support vectors (in % of the 1622 training chips) are given in Figure 1. There are displayed the maximums relative to the kernel parameter, i.e. the results with highest value for the weakest class.

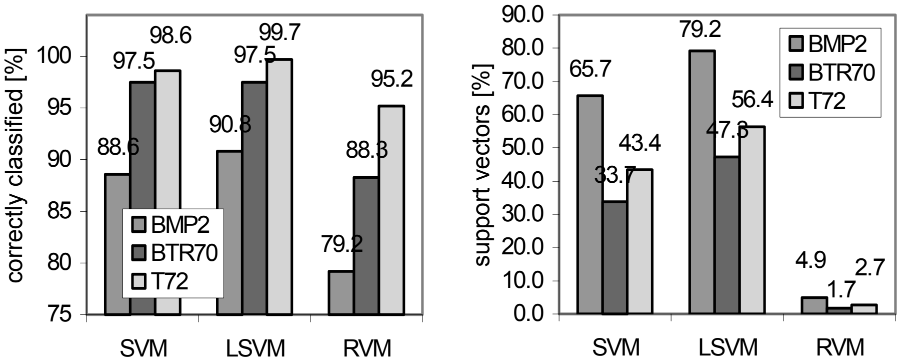

Fig. 1. Classification quality and number of SVs / RVs for the SVM, LSVM, and RVM.

The kernel machines are of quite different nature. The SVM results in a good quality using a lot of SVs. The LSVM is a little bit better using more SVs than the SVM. The RVM yields in a lower classification quality, but the number of Relevance Vectors (RVs) is a few times lower, i.e. the evaluation of the test function is a few times faster. In this example it is more than 15 times faster.

The objective should be to unify the advantages of the machines. We consider that SVM and RVM use the same test function, a linear combination of basis functions:

$$y(x;w) = \sum_{i=1}^{n} w_i K(x, x_i) + w_0 \qquad (3)$$

$x_i \in \Re^m$, $i = 1,...,n$ are the training vectors. The weights $w_i \in \Re$, $i = 0,...,n$ have to be computed while training the kernel machine classifier. Taking a look at the test function, RVs or SVs are the same, i.e. the naming results from different methodologies of the underlying optimization problem. The LSVM has another test function, i.e. it is used as reference only. All kernel machine classifiers taken into consideration solve two-class problems internally, i.e. the multi-class problem is divided into two-class problems via the 1-to-rest heuristic.

4. Extended Relevance Vector Machine

To modify the training algorithm we take a look at the RVM training. The RVM is using a system matrix $\Phi \in \Re^{n \times l}$, see (4). The training process only uses this system matrix, the class labels, the l so-called hyper-parameters $\alpha_i \in \Re$, and the l weights $w_i \in \Re$ for the test function.

$$\Phi = \left[\begin{array}{c:ccc} 1 & K(x_1,x_1) & \cdots & K(x_1,x_{l-1}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K(x_n,x_1) & \cdots & K(x_n,x_{l-1}) \end{array}\right] \in \Re^{n \times l} . \quad (4)$$

At the beginning of the training it holds $l = n + 1$. During the iteration basis functions like $K(x, x_{i_0})$ have to be eliminated to guarantee a numerically stable algorithm. The necessity of this elimination is indicated by the respective hyper-parameter α_{i_0} that is growing towards infinity. On the other hand a growing hyper-parameter results in $w_{i_0} \to 0$, i.e. the basis function has no influence on the value of the test function (3), so it can be eliminated. The update of the system is done by eliminating the corresponding column of the system matrix Φ, the corresponding α_{i_0}, and w_{i_0}. Further it is set $l = l - 1$. For more details see [6, 7].

The columns of Φ correspond to basis functions. The rows of Φ are not eliminated during the training. They correspond to the training data, i.e. the optimization process is internally controlled by the classification quality for this training dataset.

This described structure of the RVM makes it possible to choose the basis functions and the RVM training dataset independently of each other.

Therefore a first shot is done to yield an upper limit of high classification quality and low computational effort for the considered family of kernel machines. We have chosen the training data for the basis functions and the union of training and test dataset for the RVM training dataset. This setup is done to determine an upper limit only. We have set $l = n_{train} + 1$, $n = n_{train} + n_{test}$, and have calculated the system matrix Φ by evaluating the RBF kernel. The cross-classification result of this experimental RVM (RVME) for the MSTAR dataset is given in Table 2. The percent rates are relative to 1365 test or 1622 training chips respectively.

Table 2. Classification result of RVME with RBF kernel parameter $\sigma = 60$

Class	Correct [%]	RVs [%]
BMP2	99.8	6.0
BTR70	100.0	2.1
T72	99.3	3.8

It is possible to get a kernel classifier for basis functions relative to the MSTAR training dataset that does a quasi 100% classification of the test dataset and that uses all over 11.9% RVs only. This result is a possible limit. It strongly depends on the local correlation between training and test dataset in vector space F. For realistic considerations we do not have any knowledge about the test dataset. But such modification of the RVM is an example of a universal method.

5. RVMG – Relevance Vector Machine with Generator

The advantage of the SVM versus the RVM is its good classification quality. This can be explained by maximizing the margin between classes and the discriminating hyperplane in vector space F , the kernel generating function is mapping into. A description of this is the maximization of spheres of same size around all training points. These spheres have to permit a linear discrimination with respect to the classes. The RVM does not respect a neighborhood of data points in contrast to SVM.

Many generators are possible to build a discrete neighborhood of training points. One type produces additional points around all original points, e.g. within a given distance. Other ones use families of geometrical transformations to the original data. Such generators often have several disadvantages. In contrast to this we define objectives of the generator which has to be defined:

- No blurring of class boundaries, so the overlapping of classes does not grow.
- It should not increase the class density of irrelevant inner regions of classes.
- Producing a manageable number of points that does not make the training impossible on real computers.
- A variable distance between original and generated points with respect to the underlying structure of classes.

With the method described in the previous section it is possible to extend the training dataset by such generated data. What kind of generator we use and how to overcome the problem of mapping into F is described as follows.

5.1. Generator – Linear Case

The proposed generator computes possible boundary or near boundary points in the first class and their counterparts in the second class, and the other way. Some of these pairs are rejected if they are not the nearest neighbors of their mean point.

$$\begin{aligned} P_0 = \{(x_i, x_j) \in C_A \times C_B \mid (x_i \text{ is NN in } C_A \text{ of } x_j \vee x_j \text{ is NN in } C_B \text{ of } x_i) \\ \wedge\, x_i \text{ is NN in } C_A \text{ of } m_{ij} \wedge x_j \text{ is NN in } C_B \text{ of } m_{ij}\} \end{aligned} . \qquad \textbf{(5)}$$

with the two training classes C_A and C_B , $m_{ij} = (x_i + x_j)/2$. NN is the Nearest Neighbor with respect to $\|.\|_2$. Additionally it holds $|P_0| \le n_{train}$.

The generator points are defined using the parameter $0 < \lambda < 0.5$ as follows:

$$\begin{aligned} G_{A,\lambda} &= \{(1-\lambda)x_i + \lambda x_j \mid (x_i, x_j) \in P_0\} \\ G_{B,\lambda} &= \{\lambda x_i + (1-\lambda)x_j \mid (x_i, x_j) \in P_0\}. \end{aligned} \qquad \textbf{(6)}$$

The number of training points $n = |C_A| + |C_B| + |G_{A,\lambda}| + |G_{B,\lambda}| \le 3n_{train}$, inclusive the generated data $G_{A,\lambda}$ and $G_{B,\lambda}$ relative to the two classes, is of moderate size.

5.2. Generator – Kernel Case

The nonlinear case is the interesting one. Therefore we use a kernel K. A kernel generating function $\Psi : X \to F$ theoretically exists that holds (2), but in practice only K is given. In the above subsection the generator is described for the linear case, i.e. it could be interpreted as a generator in vector space F that Ψ is mapping into. The following tasks have to be transferred from linear to kernel case.

NN in C_A or C_B for training points is solved in original vector space X. This is possible because we use the RBF kernel for which the Voronoi topology is invariant. A more efficient solution is to calculate differences of training points using elements of Φ as follows:

$$\| \Psi(x_i) - \Psi(x_j) \|_2^2 = K(x_i, x_i) - 2K(x_i, x_j) + K(x_j, x_j) \tag{7}$$

Secondly the NN in C_A or C_B of mean value $(\Psi(x_i) + \Psi(x_j))/2$ could not be computed directly. But for the NN we have to calculate the distances between mean values and training points. It could be easily shown:

$$\begin{aligned} \| (\Psi(x_i) + \Psi(x_j))/2 - \Psi(x_k) \|_2^2 \; = \; & (K(x_i, x_i) + K(x_j, x_j))/4 \\ & + K(x_k, x_k) + K(x_i, x_j)/2 - K(x_i, x_k) - K(x_j, x_k) \end{aligned} . \tag{8}$$

Thirdly the generated points have to be used as extra training data for RVM. Basis functions are only relative to the original training points. Therefore a generated point is needed to calculate elements of the corresponding new row s of Φ.

$$\Phi_{s,k} =< (1-\lambda)\Psi(x_i) + \lambda\Psi(x_j), \Psi(x_k) >= (1-\lambda)K(x_i, x_k) + \lambda K(x_j, x_k) . \tag{9}$$

Equations (7) and (8) allow to generate $G_{A,\lambda}$ and $G_{B,\lambda}$. These generated points are not processed directly but their corresponding rows of Φ are determined by equation (9). No blurring of class boundaries is done because the generated points are elements of Voronoi cells in F corresponding to same classes. The number of training points inclusive generated data is of manageable size $n = | C_A | + | C_B | + | G_{A,\lambda} | + | G_{B,\lambda} | \leq 3n_{train}$. No other than boundary regions are fortified. The relative strength (distance to boundaries) is controlled by $0 < \lambda < 0.5$.

6. Experimental Results

The experimental setup of the MSTAR dataset has been described in section 2. Several tests have been done with the described kernel machines. The results are given in Fig. 2. To yield appropriate results the machines have been parameterized with different kernel parameter values (of a small set): SVM - $\sigma = 80$, LSVM - $\sigma = 40$, RVM - $\sigma = 60$, RVME - $\sigma = 60$, RVMG - $\sigma = 40$.

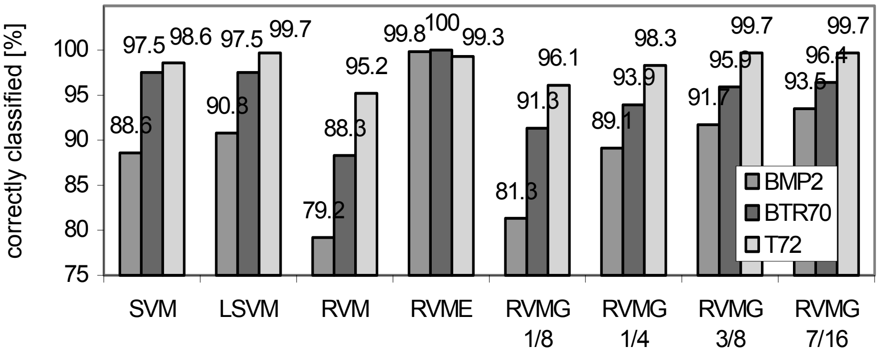

Fig. 2. Classification quality of the tested kernel classifiers (in % of the 1365 test chips).

The RVMG has been tested with several values of control parameter $0 < \lambda < 0.5$. The greater this parameter is, the higher is the classification quality, and the higher is the number of RVs. The number of SVs / RVs in % is given in Fig. 3.

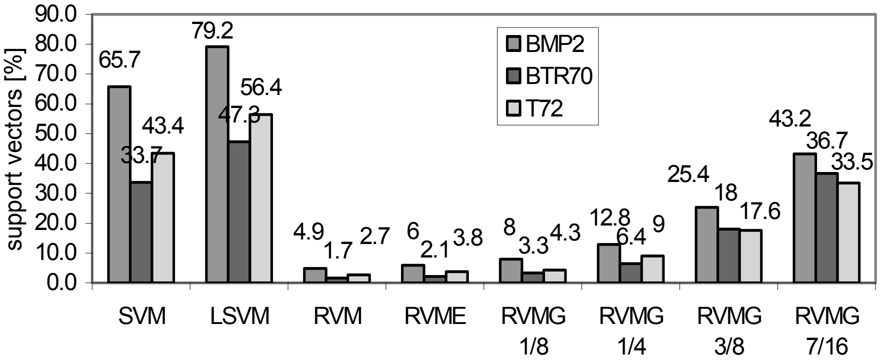

Fig. 3. Number of SVs / RVs of the tested kernel classifiers (in % of the 1622 training chips).

The spectrum varies from a 15 times faster machine than the RVM with 10% lower quality than the SVM, goes to a 5 times faster and equal quality machine – the RVMG ($\lambda = 1/4$), and ends with the RVMG ($\lambda = 7/16$), a machine a little bit faster than the SVM and of better quality than the LSVM. The quality of the RVME could not be reached any more. But for its training it uses the knowledge about the test data.

The greater the parameter λ has been chosen, the harder the training, e.g. the training of the RVMG with $\lambda = 1/4$ has been done in 196 minutes, but with $\lambda = 7/16$ it takes 105 hours. Fortunately the time of training has no influence on the evaluation of the test function. In all cases the cross-classification of the MSTAR test dataset only takes a few minutes on a computer with AMD 1800+.

RVMG is a controllable machine that is adjustable for a higher classification speed while decreasing the classification quality. And it is feasible to reach the quality of the LSVM with lower computational effort.

7. Conclusion

With the described new approach it is possible to construct a kernel machine classifier with indirectly controlled trade-off between speed and quality. For this a single parameter controls the strength of fortification of class boundaries, i.e. the robustness of the original class structure for the classification training. Class boundaries are respected, i.e. no additional class overlapping is produced. The generated extended training dataset is of moderate size. Therefore the proposed RVMG is prepared for a wide range of applications.

Further investigations will be done with other datasets. A preprocessing of data should be taken into consideration. Other generator approaches are possible using the described method and the choice of different kernels should be discussed.

Acknowledgements

The author wishes to thank U. Thönnessen FGAN-FOM for his permanent advice and helpful discussions. Moreover many thanks to A. Ebert FGAN-FOM for his technical assistance in implementing and testing the software system.

References

1. Müjdat Çetin, W. Clem Karl, and David A. Castañon, "Analysis of the Impact of Feature-Enhanced SAR Imaging on ATR Performance", Procedings of SPIE - Algorithms for SAR Imagery IX, E.G. Zelnio (ed.), 4727, 134-145, Orlando, Florida, 2002
2. N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press, 2000
3. O. L. Mangasarian, D. R. Musicant, "Lagrangian Support Vector Machines", Journal of Machine Learning Research, 1, 161-177, 2001
4. W. Middelmann, U. Thönnessen, "Classification in High-Resolution SAR data", Proceedings of SPIE - Algorithms for SAR Imagery, E. G. Zelnio (ed.), to be published
5. B. Schölkopf, Support Vector Learning, PhD Thesis, R. Oldenbourg Verlag, Munich, 1997
6. M. E. Tipping, "The Relevance Vector Machine", Advances in Neural Information Processing Systems, 12, S. A. Solla, T. K. Leen, K.-R. Muller (eds), MIT Press, Cambridge, MA, 2000
7. M. E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine", Journal of Machine Learning Research, 1, 211-244, 2001
8. J. Tsao, B. D. Steinberg, "Reduction of Sidelobe and Speckle Artifacts in Microwave Imaging: The CLEAN Technique", IEEE Transactions on Antennas and Propagation, 36, 543-556, 1988
9. V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998
10. MSTAR, Air Force Research Lab., Model Based Vision Lab., Sensor Data Management System, http://www.mbvlab.wpafb.af.mil/public/sdms/datasets/mstar/overview.htm